

Spring Boot Mongo

MongoDB 简介

MongoDB 是专门为可扩展性、高性能和高可用性而设计的数据库，它可以从单服务器部署扩展到大型、复杂的多数据中心架构。利用内存计算的优势，MongoDB 能够提供高性能的数据读写操作。MongoDB 的本地复制和自动故障转移功能使应用程序具有企业级的可靠性和操作灵活性。

- MongoDB 相关概念

SQL 术语/概念	MongoDB 术语/概念	解释/说明
DataBase	DataBase	数据库
Table	Collection	数据库表/集合
Row	Document	数据记录行/文档
Column	Field	数据字段/域
index	index	索引
Table joins		表连接，MongoDB 不支持
primary key	primary key	主键，MongoDB 自动将 _id 字段设置为主键

MongoDB 和关系数据库一样有库的概念，一个 MongoDB 数据库可以有很多数据库，MongoDB 中的集合就相当于我们关系数据库中的表，文档就相当于关系数据库中的行，域就相当于关系数据库中的列，MongoDB 也支持各种索引有唯一主键，但不支持表连接查询。

Spring Boot MongoDB

MongoTemplate 的使用

MongoTemplate 提供了非常多的操作 MongoDB 方法，它是线程安全的，可以在多线程的情况下使用。

MongoTemplate 实现了 MongoOperations 接口, 此接口定义了众多的操作方法如 find、findAndModify、findOne、insert、remove、save、update and updateMulti 等。它转换

domain object 为 DBObjcet, 并提供了 Query、Criteria and Update 等流式 API。

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-mongodb</artifactId>
  </dependency>
</dependencies>
```

- application.properties 中添加配置

```
spring.data.mongodb.uri=mongodb://name:pass@localhost:27017/test
```

多个 IP 集群可以采用以下配置：

```
spring.data.mongodb.uri=mongodb://user:pwd@ip1:port1,ip2:port2/database
```

当然了密码和用户名如果没有设置可以不用添加，像这样：

```
spring.data.mongodb.uri=mongodb://localhost:27017/test
```

- 创建数据实体

```
public class User implements Serializable {
    private static final long serialVersionUID = -3258839839160856613L;
    private Long id;
    private String userName;
    private String passWord;
}
```

- 对实体进行增删改查操作

```
@Component
public class UserRepositoryImpl implements UserRepository {

    @Autowired
    private MongoTemplate mongoTemplate;

}
```

- 添加数据:

```
@Override
public void saveUser(UserEntity user) {
    mongoTemplate.save(user);
}
```

save 方法中会进行判断，如果对象包含了 ID 信息就会进行更新，如果没有包含 ID 信息就自动保存。

- 更新

```
@Override
public long updateUser(User user) {
    Query query=new Query(Criteria.where("id").is(user.getId()));
    Update update= new Update().set("userName", user.getUserName()).set("password",
user.getPassword());
    //更新查询返回结果集的第一条
    UpdateResult result =mongoTemplate.updateFirst(query,update,User.class);
    //更新查询返回结果集的所有
    // mongoTemplate.updateMulti(query,update,UserEntity.class);
    if(result!=null)
        return result.getMatchedCount();
    else
        return 0;
}
```

- 删除对象:

```
@Override
public void deleteUserById(Long id) {
    Query query=new Query(Criteria.where("id").is(id));
    mongoTemplate.remove(query,User.class);
}
```

MongoRepository

MongoRepository 继承于 PagingAndSortingRepository，再往上就是

CrudRepository,MongoRepository 和前面 JPA、Elasticsearch 的使用比较类似，都是 Spring Data 家族的产品，最终使用方法也就和 JPA、ElasticSearch 的使用方式类似，可以根据方法名自动生成 SQL 来查询。

```
public interface UserRepository extends MongoRepository<UserEntity, Long> {  
    UserEntity findByUserName(String userName);  
}
```

- 新建 UserRepository 需要继承 MongoRepository，这样就可直接使用 MongoRepository 的内置方法。
- 两种方式都可以方便地操作 MongoDB 数据库，MongoTemplate 模式比较灵活可以根据自己使用情况进行组装，MongoRepository 的使用方式更简单，因为继承了 Spring Data，所以默认实现了很多基础的增、删、改、查功能，业务直接拿来使用即可。简单日常的使用推荐 MongoRepository，简单方便利于项目快速开发，复杂多变的查询推荐使用 MongoTemplate 自行进行封装。